

CLAIMS

We claim:

1. A method for a first process running on a computing device to communicate with a second process, the method comprising:
 - 5 creating a process table on the computing device;
 - rendering the process table accessible to the first process;
 - associating a Universally Unique Identifier (UUID) with the second process;
 - creating an entry for the second process in the process table;
 - 10 associating the UUID of the second process with the process entry for the second process in the process table;
 - specifying a communications task to perform; and
 - using the UUID of the second process to specify that the communications task be performed with respect to the second process.
 - 15
2. The method of claim 1 wherein creating a process table comprises creating the process table as shared memory on the computing device.
3. The method of claim 1 further comprising:
 - 20 coordinating access to the process table and to the process entry for the second process via software locks.
4. The method of claim 1 wherein specifying a communications task to perform comprises specifying monitoring a process and wherein the method further comprises:
 - 25 writing status information about the second process into the process entry for the second process; and
 - retrieving the status information about the second process by using the UUID of the second process to access the process entry for the second process in
 - 30 the process table.

5. The method of claim 4 wherein writing status information comprises periodically writing a heartbeat update time and wherein the method further comprises:
 - comparing the heartbeat update time in the status information to the current time; and
 - 5 determining if the second process is running based on the comparing of the times.
6. The method of claim 1 wherein specifying a communications task to perform comprises specifying requesting information from a process and wherein the method further comprises:
 - 10 specifying a type of information requested; and
 - returning the information requested to the first process.
7. The method of claim 6 wherein the type of information requested is selected from the set: log output, console output.
 - 15
8. The method of claim 6 further comprising:
 - specifying a period of time during which to return the information requested; and
 - 20 wherein returning comprises returning the information requested during the specified period of time.
9. The method of claim 6 wherein returning comprises returning the information requested until the first process indicates that the information need no longer be returned.
 - 25
10. The method of claim 1 wherein specifying a communications task to perform comprises specifying waiting for the second process to achieve a status.
- 30 11. The method of claim 10 wherein the status is in the set: initialized, debug_break, terminated.

12. The method of claim 1 wherein specifying a communications task to perform comprises specifying sending a signal to the second process.
- 5 13. The method of claim 12 wherein sending a signal indicates that the process should terminate.
14. The method of claim 1 further comprising:
- 10 associating a UUID with a third process;
 creating an entry for the third process in the process table;
 associating the UUID of the third process with the process entry for the
third process in the process table;
 associating the UUID of the second process with the process entry for the
third process in the process table; and
- 15 using the UUID of the second process to specify that the communications
task be performed with respect to the third process.
15. The method of claim 14 wherein the third process is a child of the second process.
- 20 16. The method of claim 15 further comprising using the UUID of the second process
to specify that the communications task be performed with respect to all
descendents of the second process.
17. The method of claim 1 wherein the second process runs on a second computing
25 device distinct from the computing device on which the first process runs.

18. The method of claim 17 further comprising:
- associating an identifier of the second computing device with the process entry for the second process in the process table;
 - creating a second process table on the second computing device;
 - 5 creating an entry for the second process in the second process table; and
 - associating the UUID of the second process with the process entry for the second process in the second process table.
19. The method of claim 18 wherein specifying a communications task to perform comprises specifying monitoring a process and wherein the method further comprises:
- 10 writing status information about the second process into the process entry for the second process in the second process table; and
 - retrieving the status information about the second process by using the
 - 15 UUID of the second process to access the process entry for the second process in the second process table.
20. A computer-readable medium having instructions for performing the method of claim 1.

21. A method for a first process running on a computing device to communicate with a second process and with a third process, the method comprising:
 - creating a process table on the computing device;
 - rendering the process table accessible to the first process;
 - 5 creating an entry for the second process in the process table;
 - creating an entry for the third process in the process table;
 - associating a group UUID with the process entry for the second process in the process table;
 - associating the group UUID with the process entry for the third process in the process table;
 - 10 specifying a communications task to perform; and
 - using the group UUID to specify that the communications task be performed with respect to the second and third processes.
- 15 22. The method of claim 21 wherein creating a process table comprises creating the process table as shared memory on the computing device.
23. The method of claim 21 further comprising:
 - coordinating access to the process table and to the process entries for the second and third processes via software locks.
 - 20
24. The method of claim 21 wherein specifying a communications task to perform comprises specifying monitoring a process and wherein the method further comprises:
 - 25 writing status information about the second process into the process entry for the second process;
 - writing status information about the third process into the process entry for the third process; and
 - retrieving the status information about the second and third processes by
 - 30 using the group UUID to access the process entries for the second and third processes in the process table.

25. The method of claim 24 wherein writing status information comprises periodically writing a heartbeat update time and wherein the method further comprises:
- 5 comparing the heartbeat update times in the status information to the current time; and
- determining if the second and third processes are running based on the comparing of the times.
26. The method of claim 21 wherein specifying a communications task to perform
- 10 comprises specifying requesting information from a process and wherein the method further comprises:
- specifying a type of information requested; and
- returning the information requested to the first process.
- 15 27. The method of claim 26 wherein the type of information requested is selected from the set: log output, console output.
28. The method of claim 26 further comprising:
- 20 specifying a period of time during which to return the information requested; and
- wherein returning comprises returning the information requested during the specified period of time.
29. The method of claim 26 wherein returning comprises returning the information
- 25 requested until the first process indicates that the information need no longer be returned.
30. The method of claim 21 wherein specifying a communications task to perform
- 30 comprises specifying waiting for the second process to achieve a status and the third process to achieve the status.

31. The method of claim 30 wherein the status is in the set: initialized, debug_break, terminated.
- 5 32. The method of claim 21 wherein specifying a communications task to perform comprises specifying waiting for the second process or the third process to achieve a status.
33. The method of claim 32 wherein the status is in the set: initialized, debug_break, terminated.
- 10 34. The method of claim 21 wherein specifying a communications task to perform comprises specifying sending a signal to the second and the third processes.
- 15 35. The method of claim 34 wherein sending a signal indicates that a process should terminate.
36. The method of claim 21 wherein the second process runs on a second computing device distinct from the computing device on which the first process runs.
- 20 37. The method of claim 36 further comprising:
 associating an identifier of the second computing device with the process entry for the second process in the process table;
 creating a second process table on the second computing device;
 creating an entry for the second process in the second process table; and
25 associating the group UUID with the process entry for the second process in the second process table.

38. The method of claim 37 wherein specifying a communications task to perform comprises specifying monitoring a process and wherein the method further comprises:
- 5 writing status information about the second process into the process entry for the second process in the second process table; and
- retrieving the status information about the second process by using the group UUID to access the process entry for the second process in the second process table.
- 10 39. A computer-readable medium having instructions for performing the method of claim 21.
40. A computer-readable medium having stored thereon a data structure, the data structure comprising:
- 15 a first data field containing data representing a UUID associated with a process; and
- a second data field containing data representing a process identifier associated with the process by an operating system.
- 20 41. The data structure of claim 40 further comprising:
- a third data field comprising data representing a UUID associated with a parent process of the process.
42. The data structure of claim 40 further comprising:
- 25 a third data field comprising data representing a UUID associated with a group comprising the process.

43. The data structure of claim 40 further comprising:
a third data field comprising data representing a time of creation of the process;
a fourth data field comprising data representing the most recent time that
5 the process logged a heartbeat; and
a fifth data field comprising data representing a type of the process.
44. The data structure of claim 40 further comprising:
a third data field comprising data representing an identity of a computing
10 device on which the data structure resides; and
a fourth data field comprising data representing an identity of a computing
device on which the process runs.
45. The data structure of claim 44 wherein the identities of the computing devices are
15 represented by data in the set: name, IP address.
46. A computer-readable medium having stored thereon a data structure, the data
structure comprising:
a first data field containing data representing a type of the new process;
20 a second data field containing data representing a UUID; and
a third data field containing data representing a command line to execute
to initiate the process.
47. The data structure of claim 46 wherein the UUID is a NIL UUID.
25
48. The data structure of claim 46 further comprising:
a fourth data field comprising data representing a username to use when
creating the process; and
a fifth data field comprising data representing a password to use when
30 creating the process.

49. The data structure of claim 46 further comprising:
a fourth data field comprising data representing a directory in which to execute the process.
- 5 50. The data structure of claim 46 further comprising:
a fourth data field comprising data representing a UUID of a parent of the process.
- 10 51. The data structure of claim 46 further comprising:
a fourth data field comprising data representing a UUID of a group comprising the new process.
- 15 52. The data structure of claim 46 further comprising:
a fourth data field comprising data representing a computing device on which the process will run.
53. The data structure of claim 52 wherein the data representing the computing device are in the set: name, IP address.
- 20 54. A method for requesting an operating system to create a new process, the method comprising:
issuing a create process call with input parameters comprising a type of the new process, a UUID, and a command line to execute to initiate the new process;
receiving, by the operating system, the create process call, parsing the call to retrieve the input parameters, and executing the command line; and
25 issuing, by the operating system, an acknowledgement of the create process call with acknowledgement parameters comprising a handle to information related to the new process.

55. The method of claim 54 wherein issuing a create process call comprises issuing a call with a NIL UUID in the input parameters and further comprising:
associating, by the operating system, a non-NIL UUID with the new process.
- 5
56. A computer-readable medium having instructions for performing the method of claim 54.
57. A method for waiting for multiple processes to achieve a status, the method comprising:
issuing a wait for multiple processes call with input parameters comprising a list of processes for which to wait, the status to achieve, and a timeout period;
receiving the wait for multiple processes call, parsing the call to retrieve the input parameters, and periodically checking a heartbeat of processes in the list of processes until all processes reach the status or reach a default status or until the timeout period is passed; and
issuing an acknowledgement of the wait for multiple processes call with acknowledgement parameters comprising the status of the processes.
- 10
- 15
- 20
58. A computer-readable medium having instructions for performing the method of claim 57.

59. A method for waiting for one of multiple processes to achieve a status, the method comprising:
- issuing a wait for multiple processes call with input parameters comprising a list of processes for which to wait, the status to achieve, and a timeout period;
- 5 receiving the wait for multiple processes call, parsing the call to retrieve the input parameters, and periodically checking a heartbeat of processes in the list of processes until any one of the processes reaches the status or until all of the processes reach a default status or until the timeout period is passed; and
- issuing an acknowledgement of the wait for multiple processes call with
- 10 acknowledgement parameters comprising the status of the processes.
60. A computer-readable medium having instructions for performing the method of claim 59.
- 15 61. A method for retrieving information about a process, the method comprising:
- issuing a get process information call with input parameters comprising an identification of the process, a list of types of information to retrieve, an identification of a computing device from which to retrieve the types of information, and a resolve remote flag indicating whether the computing device
- 20 should query a remote computing device for the types of information if the process runs on a remote computing device;
- receiving the get process information call, parsing the call to retrieve the input parameters, and, if the process runs on a remote computing device and if the resolve remote flag is set to TRUE, then requesting the information from the
- 25 remote computing device; and
- issuing an acknowledgement of the get process information call with acknowledgement parameters comprising the types of information requested.
62. A computer-readable medium having instructions for performing the method of
- 30 claim 61.

63. A method for retrieving output from a process, the method comprising:

issuing a get process output call with input parameters comprising an identification of the process, a type of output to retrieve, and a callback function to retrieve the output asynchronously;

5 receiving the get process output call, parsing the call to retrieve the input parameters, and initiating the callback function to retrieve the output; and

issuing an acknowledgement of the get process output call with acknowledgement parameters comprising a flag indicating whether the callback function was successfully initiated.

10

64. The method of claim 63 wherein the issuing a get process output call comprises issuing a call to retrieve output in the set: log output, console output.

65. A computer-readable medium having instructions for performing the method of claim 64.

15